

**REMARKS**

Reconsideration of the application as amended herein is respectfully requested. Claim 1 has been amended. Claims 1-7 and 10 are pending.

**Claim Objections**

Claim 1 has been objected to because it contained a typographical error. Applicant has amended claim 1 to correct this typographical error. Applicant apologizes for any inconvenience this may have caused.

**Title Objection**

The Office Action objects to the Title. Applicant notes that this application was filed with the correct spelling of “Design” in the Title. Thus, Applicant respectfully request that the Patent and Trademark Office correct their records to accurately reflect that the title of this application is “Hardware-Assisted Design Verification System Using A Packet-Based Protocol Logic Synthesized For Efficient Data Loading And Unloading”.

**Claim Rejections - 35 U.S.C. § 102(b)**

The Office Action has rejected claims 1-7 under 35 U.S.C. §102(b) as being anticipated by Sample. Applicant respectfully traverses this rejection. As an initial matter, Applicant respectfully notes that Sample is not directed to increasing communication bandwidth between a *workstation* and a *functional verification system*, as is required by the claims. The Office Action points to Col. 3, lines 45-50, as support for such a teaching. Applicant respectfully notes that this is not correct. In particular, Sample teaches that placing processors in communication with

the logic chips making up the emulator such that the logic chips calculate events and the like will “*minimize*” overhead of event transfer. In contrast, the present claims are directed to a system that “*increases*” bandwidth between a functional verification system (which is comprised of multiple chips) and a workstation, which is external to the functional verification system. Plainly, “minimizing” overhead is not the same thing as “increasing” bandwidth. In fact, the claims require synthesizing accessibility logic into the user’s logic design, which, rather than minimizing overhead, actually increases overhead.

In addition, the Office Action erroneously states that Sample’s “programmable interconnect” corresponds to the “accessibility logic” required by claim 1. Sample discloses that the programmable interconnect in his system is as follows:

The system includes one or more reconfigurable logic components which may be programmable gate array ("FPGA") devices 10 interconnected using the programmable interconnect 12. The interconnect 12 can be programmed to create an arbitrary connection between any number of inputs or outputs of the devices connected to it.

Sample, Col. 4, lines 60-65. This quote makes it clear that Sample’s programmable interconnect is structurally and functionally different than the accessibility logic required by claim 1.

As to function, this quote makes it clear that Sample’s “programmable interconnect” is used to arbitrarily interconnect the inputs and outputs of the FPGA devices. In contrast, the “accessibility logic” required by claim 1 is used to create access ports to the memories and registers *in the user’s logic design*. As the specification of the present application makes clear, the “user’s logic design” is the design that will be verified in the functional verification system. See, e.g., Paragraph 19 of the present application.

Structurally, Sample’s programmable interconnect is a part of the emulator’s hardware, and is programmed to allow the FPGAs to be interconnected. In contrast, claim 1 requires that

the accessibility logic be *synthesized into the user's design*. In claim 1, it is the user's design (supplemented by the accessibility logic) that will be programmed into the functional verification system. The accessibility logic has nothing to do with interconnecting any programmable logic chips that might make up the functional verification system. Rather, as discussed, the accessibility logic is synthesized into the user's design, which is not part of the hardware making up the functional verification system of claim 1.

In addition, in the context of claim 1, the Office Action is wrong when it equates behavioral fragments with the "user's design", as it does on pages 4-5. Behavioral fragments relate to behavioral representations of a design that will be simulated in the processor, as selected by the FPGA. In the claim language quoted by the Office Action, the "user's design" is referred to in the context of the "accessibility logic" that will be synthesized into the user's design. Thus, equating behavioral fragments, which are not products of synthesis, with the accessibility logic is not correct.

The Office Action is also wrong when equating Sample's use of a system controller 26 to download configuration data into the FPGAs and executable data into the random access memories. Claim 1 requires that the accessibility logic, which comprises access ports, facilitate writing data into and reading data from memories and register's in the user's design. The sections from Sample quoted by the Office Action make it clear that no such features are disclosed. First, the "system controller 26", as seen in Figure 1 of Sample, is external to the FPGAs and programmable interconnect. In fact, the system controller 26 is part of the Sample emulation hardware ("The system controller 26 is implemented using a commercial embedded controller board..." Col. 5, lines 62-64), and *is not synthesized into the user's design*, as is required by claim 1. Moreover, the configuration data downloaded into the FPGAs by the

system controller 26 is in fact the data used to program the FPGAs. This configuration data is *not* data to be written into and read from memories and registers *in the user's design*.

Likewise, the system controller 26 downloads “executable data” into the random access memory devices 20 that form hardware components of Sample’s emulator. This is not what is required by claim 1, which states that the access ports write data to and read data from the workstation into memories and registers *in the user's design*.

As is seen from this, claim 1 is distinguishable from Sample and is therefore in condition for allowance. Moreover, claims 2-5 are dependent upon claim 1, meaning they are in condition for allowance as well.

Turning now to claim 6, Applicant respectfully submits that Sample does not disclose synthesizing protocol logic *into the target logic design*. The Office Action points to Sample’s discussion of token ring protocols. However, all Sample says is that the daisy chain 32 used to interconnect FPGAs is set up with a token ring protocol. This relates to the emulation hardware and has *nothing to do with the target logic design*. As discussed, Sample does not disclose anything about synthesizing any logic into the user’s design where this logic supplements the user’s logic.

Moreover, the Office Action incorrectly equates the random access memory devices 20 with the packet registers required by claim 6. As discussed, Sample’s random access memory devices 20 are physical memories that are part of the hardware used to construct the emulation system. These random access memory devices are used by the emulation hardware to store executable data downloaded into them by system controller 26. In contrast, the packet registers of claim 6 are synthesized *into the target logic design* to store data and commands sent by the

host workstation. The packet registers of claim 6 are *not* part of the functional verification system's hardware.

Likewise, the Office Action equates Sample's operation decoder 42 with claim 6's command decode logic. This is inconsistent with the Office Action's statement that the packet registers of claim 6 are taught by Sample's random access memories 20. Claim 6 requires that the command decode logic decode commands in the incoming packet register. However, as seen in Figs. 3 and 4 of Sample, the operation decoder 42 is not even connected to random access memory device 20, meaning that Sample's operation decoder 42 could not possibly decode commands since it will never receive them. The same is true regarding the Office Action's arguments regarding the write command execution logic and read command execution logic.

Finally, the Office Action's arguments equating the interface logic required by claim 6 with the FPGAs disclosed by Sample is also erroneous. Claim 6 requires that the interface logic interface the registers and memories in the target logic circuit. In contrast, all Sample teaches is that the FPGAs have the target logic circuit programmed therein. Sample does not teach anything about *synthesizing interface logic into the target logic circuit*, which is what is required by claim 6.

In sum, claim 6 is allowable over Sample. Since claim 7 is dependent upon claim 6, claim 7 is allowable as well.

#### **Allowed Subject Matter**

The Office Action has indicated that claim 10 is in condition for allowance.

**CONCLUSION**

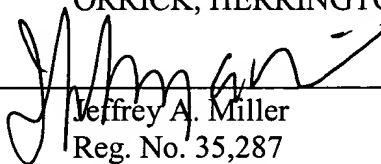
In view of the foregoing, Applicant respectfully submits that the present application is in condition for allowance, which is respectfully requested. If the Examiner believes that a telephone conference would be useful in moving the application forward to allowance, the Examiner is encouraged to contact the undersigned at (650) 614-7400. If additional fees are needed, the Office is authorized to charge Deposit Account No. 15-0665.

Respectfully submitted,

ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: July 13, 2005

By:

  
\_\_\_\_\_  
Jeffrey A. Miller  
Reg. No. 35,287

Four Park Plaza, Suite 1600  
Irvine, California 92614-2558  
(650) 614-7660